Stabilizing Real-World Visual Active Tracking with Action-Smooth Test-Time Adaptation

Haowei Sun[1⋆], **Shiteng Zhang**[1⋆], **Jinwu Hu**[1,2⋆], **Kaining Chen**[1], and **Mingkui Tan**[1,3⋆⋆]

[1] South China University of Technology
[2] Pazhou Laboratory
[3] Key Laboratory of Big Data and Intelligent Robot, Ministry of Education

Abstract. Visual Active Tracking (VAT) requires an embodied agent to control cameras to follow a designated target, which is essential for applications like robot navigation and security monitoring. However, transferring VAT trackers to the real world faces two primary bottlenecks: (1) severe visual domain shifts between simulated environments and the real world, leading to performance collapse during deployment; and (2) target distribution shift caused by similar-looking distractors in real-world scenes. To address these, we propose **VATA**, a novel test time adaptation method with two complementary strategies. First, we introduce a **Critic Value Maximization** strategy that leverages value signals from a critic model to filter reliable test samples, enabling unsupervised model updates to mitigate visual shift. Second, we propose an **Action Smoothness Regularization** term that exploits the continuity of real-world target motion to bound action variations, correcting irrational actions under target shift. Experiments demonstrate **VATA** achieves a 24.4% SR improvement on the EVT-Benchmark and reaches 80% TSR in 10 real-world scenarios (vs. 30% by the SOTA TTA method), providing a robust and plug-and-play solution for VAT deployment.

Keywords: Visual Active Tracking · Test-Time Adaptation · Reinforcement Learning · Embodied AI

1 Introduction

Visual Active Tracking (VAT) requires an embodied agent to actively control cameras to follow a designated target. It is widely used in real-world applications such as robot navigation [1,11,18,42] and security monitoring [10,28,37,40]. Unlike passive visual tracking [5,17,34,36,39,47], which focuses solely on bounding box regression on pre-recorded videos, VAT operates in a closed-loop cycle, adjusting camera motion in real-time based on visual feedback. Passive visual

⋆ Equal contribution. Email: sunhoward1105@gmail.com, zstbill@qq.com, fhujinwu@gmail.com
⋆⋆ Corresponding authors. Email: mingkuitan@scut.edu.cn

tracking often struggles in real-world environments where targets exhibit unpredictable motion. Consequently, VAT offers a more practical yet challenging solution for real-world tracking applications. VAT methods fall into pipeline-based and RL-based categories. While pipeline approaches separate perception and control modules, they suffer from heavy annotation and scene-specific tuning [30]. In contrast, RL-based methods unify perception and action in a single network, reducing system complexity and improving real-time inference. Therefore, this paper focuses on RL-based VAT methods.

Most VAT methods [9,20,30,33,44–46] rely on simulators [27,30,33] for training and evaluation, which struggle when deployed in the real world. This failure comes from two main issues: **1) Visual Domain Shift**. Real-world textures and lighting conditions differ drastically from idealized simulations, causing performance collapse. **2) Target Distribution Shift**. Real-world scenes contain many similar-looking objects, making the target distribution significantly more complex than in simulators and thus making it hard to distinguish targets from distractors. Since collecting annotated data for every new scenario is infeasible, we turn to Test-Time Adaptation (TTA) paradigm [23, 32, 38]. TTA aims to recover performance under domain shifts by identifying high-confidence test samples and using them to update the model in an unsupervised manner, offering a viable path for sim-to-real VAT.

Unfortunately, directly applying existing TTA methods to VAT remains challenging, partly for the following reasons: **1) Unreliable Optimization Signals.** Most existing methods use model output entropy to filter reliable test samples and update parameters via entropy minimization. However, in real-world VAT, agents often exhibit low output entropy even when tracking fails, as shown in Fig. 2. Consequently, using such signals introduces noisy samples, misleading the model update process. **2) Failure under Distractors.** Similar-looking distractors often cause the agent to track the wrong target, leading the tracker to make sudden moves towards the distractors. However, standard TTA methods lack regularization to correct such erratic motions.

To address these, we propose a TTA method for VAT called VATA, enabling stable tracking performance in the real world. First, we provide empirical evidence that VAT trackers often exhibit over-confidence during failure cases, making output entropy nearly identical for both successful tracking and target loss (Fig. 2). In contrast, critic values are highly discriminative. Based on this observation, we introduce a **Critic Value Maximization** strategy that uses critic value to filter reliable samples, guiding model updates under visual domain shift. Second, we provide a theoretical justification that since real-world target motion is continuous and bounded by maximum velocity, the variation in optimal tracker actions has an upper bound (see **Proposition** 1). This implies that any action jump exceeding this limit indicates irrational control. Based on this insight, we propose an **Action Smoothness Regularization** term to correct irrational actions under distractors, ensuring the tracker locks onto the correct target under target distribution shift. Our main contributions are as follows:

- **The First TTA Method for Real-world VAT.** We propose VATA, the first TTA method to bridge the sim-to-real gap in visual active tracking.
- **A Generic Action Regularization Term.** We introduce an action regularization term to correct erratic actions caused by distractors, which is compatible with both RL-based and IL-based VAT trackers.
- **Extensive Real-World Validation.** We present the first extensive sim-to-real evaluation of TTA in VAT, deploying VATA on a physical drone across diverse indoor and outdoor scenarios. Our method achieves an 80% tracking success rate, marking a 4× improvement over the base tracker (20%).

2 Related Work

2.1 Visual Active Tracking

Visual Active Tracking (VAT) aims to autonomously follow a specific target by controlling the motion system of the tracker based on continuous visual observations. Existing VAT methods can be broadly categorized into two paradigms: **pipeline-based** and **reinforcement learning (RL)-based** trackers. **Pipeline-based** methods [4, 21, 33, 35, 43] decouple VAT into perception and control stages. The perception stage typically employs video-based passive tracking models [2,3,5,17,29,34,39,47] that take an image and an initial target bounding box as input to output the current target location. The output bounding box then serves as input for a controller (e.g., PID [22]) to keep the target centered in the view. However, the dependence on initialization with a ground-truth bounding box limits their applicability in real-world tasks. To address this, FollowAnything [21] leverages visual foundation models (e.g., SAM [14], DINOv2 [25]) to extract features from a reference image for target matching during tracking.

Furthermore, targets are frequently occluded by obstacles in real-world scenarios, yet simple PID controllers often fail to navigate around them, causing permanent target loss. To address this, TrackVLA [33] incorporates a diffusion-based planner to generate future waypoints, enabling effective target recovery under occlusion. With the advances in Multimodal Large Language Models (MLLMs), several methods integrate MLLMs into VAT to enhance tracker performance [33,35]. Among them, TrackVLA [33] employs Vicuna-7B [6] as a perception backbone combined with a diffusion-based planning module, fine-tuning the entire framework end-to-end via imitation learning on large-scale datasets. Besides, LLM-Assistant [35] uses off-the-shelf VLMs (e.g., GPT-4o) for environment understanding to assist the tracker in high-level planning.

In comparison, **RL-based** VAT trackers [8, 9, 20, 30, 44–46] seek to integrate visual tracking and control within a unified framework. By mapping visual inputs directly to actions through a single network, these methods eliminate the need for separate tuning of tracking and control modules. SARL [20] pioneered the use of RL for end-to-end VAT. Subsequent works like AD-VAT+ [44] formulate VAT as an adversarial pursuit problem to enhance agent robustness, while EVT [46] introduces offline RL to improve training efficiency. Beyond ground-based tracking, GC-VAT [30] and D-VAT [9] extend RL to air-to-ground and drone-to-drone

tracking tasks, respectively. Compared to pipeline-based approaches, RL-based methods offer advantages in lightweight models and real-time inference but face significant challenges in sim-to-real policy transfer. Consequently, this paper focuses on RL-based VAT trackers for robust real-world deployment.

2.2 Test-Time Adaptation

Test-Time Adaptation (TTA) aims to address the domain shift between training and test scenarios by updating the model online with test data in an unsupervised paradigm. Compared to methods that require annotated data from the target scenario, TTA relies solely on incoming test streams, offering superior efficiency.

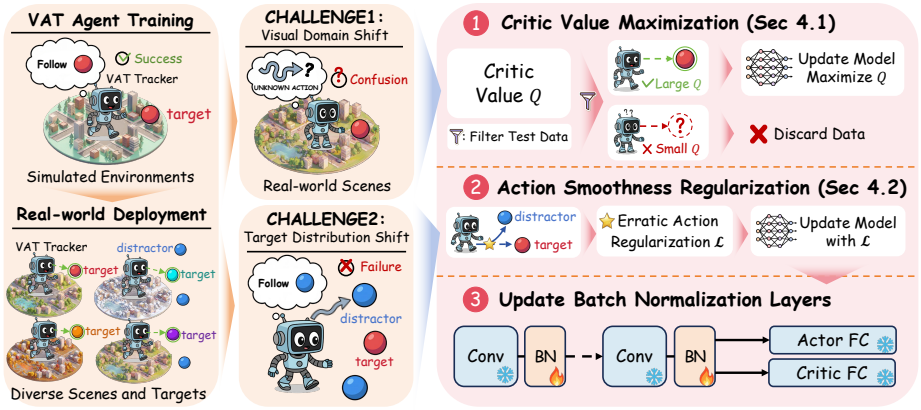
Current TTA methods [23,24,32,38] primarily leverage prediction entropy as a metric, optimizing the model via entropy minimization. For instance, EATA [23] employs entropy to select high-confidence samples for updates while introducing a Fisher regularization term to mitigate catastrophic forgetting. Furthermore, TARL [38] extends TTA to Offline RL, addressing the domain shift between offline datasets and online environments via entropy minimization, and incorporates KL-divergence regularization to prevent overfitting. Beyond perception tasks, recent works have integrated TTA into embodied AI tasks [13, 15, 31]. For example, FS-TTA [13] proposes a joint decomposition-accumulation strategy to balance update stability and environmental adaptation in online Vision-and-Language Navigation (VLN). Similarly, EAM [31] introduces an auxiliary decision model and a sample replay mechanism to alleviate catastrophic forgetting and unstable parameter updates in source-free online VLN. However, none of these methods truly address the sim-to-real gap. They are typically evaluated only in simulators under out-of-distribution settings where the domain gap remains limited. In contrast, our work leverages TTA specifically to overcome the sim-to-real challenges in visual active tracking, with comprehensive evaluations conducted in real-world environments.

3 Task Definition

Visual Active Tracking (VAT) aims to autonomously follow a specific target in dynamic 3D environments using egocentric images. We formulate this task as a Markov Decision Process (MDP): $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$. In this representation, $\mathcal{S}$ denotes the observation space, $\mathcal{A}$ represents the action space, and $\mathcal{R}$ is the reward function. At timestep t, the tracker receives state $s_t \in \mathcal{S}$ from the environment and performs an action $a_t \in \mathcal{A}$ to adjust its camera pose. Then, the simulator transitions to the next state $s_{t+1} = \mathcal{P}(s_t, a_t)$ and calculates the reward $r_t = \mathcal{R}(s_t, a_t)$ for current timestep. The details of the MDP are as follows:

Observation Space $\boldsymbol{\mathcal{S}}$. At each time step t, the agent receives an RGB image $\mathcal{I}_t$ from simulators or real robots.

Action Space $\boldsymbol{\mathcal{A}}$. Our tracker supports discrete and continuous control tasks. For discrete tasks (e.g., DAT [30]), $\mathcal{A}$ consists of 7 high-level commands:



(a) Challenges of real-world VAT deployment (b) Adaptation strategies of the proposed VATA method

Fig. 1: Overview of the proposed VATA method. (a) Deploying a simulator-trained VAT tracker in the real world faces two key challenges, visual domain shift confuses the tracker's perception, while target distribution shift causes it to track the wrong object. (b) At runtime, VATA selects reliable test samples via critic values for model updates, and applies action smoothness regularization to correct erratic actions caused by distractors. All model updates are applied only to the batch normalization layers.

forward, *backward*, *leftward*, *rightward*, *turn left*, *turn right*, and *stop*. For continuous tasks (e.g., UnrealCV [27]), $\mathcal{A}$ includes 3-dimensional low-level velocity commands $[v_x, v_y, \omega]^T$, representing linear velocities and yaw rate.

Reinforcement Learning-based VAT Tracker. We formulate the tracker as an embodied agent with an Actor-Critic architecture. The actor model $\pi_{\Theta_a} : \mathcal{S} \rightarrow \mathcal{A}$ maps observations to actions: $a_t = \pi_{\Theta_a}(s_t)$, while the critic model Q_{Θ_c} evaluates expected values based on current state s_t or state-action pair (s_t, a_t).

Success Criterion. A tracking episode is defined as successful if the agent maintains the target centered in the image frame for a long duration.

Test-Time Adaptation on VAT. Test-time adaptation (TTA) aims at boosting the out-of-distribution performance by updating model on test data only. Specifically, given an observation s_t, we need to update the actor model π_{Θ_a} to predict more accurate action a_t. To achieve this, it is necessary to formulate unsupervised objectives on test samples and update the model by:

$$\min_{\tilde{\Theta}_a} \mathcal{L}(s_t; \Theta_a), \tag{1}$$

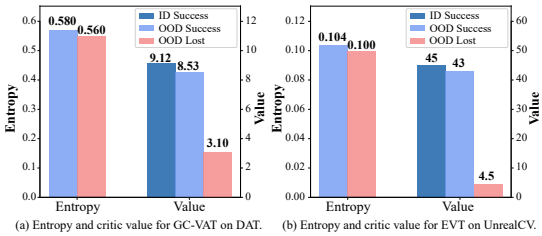
where $\tilde{\Theta}_a \subseteq \Theta_a$ denotes the adaptable parameters during TTA, typically the parameters of batch normalization layers.

4 Visual Active Tracking with Test Time Adaptation

Visual active tracking in the real world is extremely challenging, primarily because practical environments exhibit severe visual domain shifts and substantial

Algorithm 1 The pipeline of proposed VATA

Require: Current observation s_t, the trained actor model $\pi_{\Theta_a}(\cdot)$, the trained critic model $Q_{\Theta_c}(\cdot)$, historical action buffer $\mathcal{H}_t = \{a_{t-i}\}_{i=1}^{K}$
Ensure: Updated actor model $\pi_{\Theta'_a}(\cdot)$, the predicted action a_t
1: Obtain action: $a_t = \pi_{\Theta_a}(s_t)$
2: Obtain critic value: $Q_t = Q_{\Theta_c}(s_t, a_t)$
3: Compute maximization value loss $\mathcal{L}_{val}$ with Q_t via Eq. (2) or Eq. (3)
4: Compute action smoothness loss $\mathcal{L}_{smooth}$ with $a_t, \mathcal{H}_t$ via Eq. (5) and Eq. (6)
5: Compute total loss $\mathcal{L}_{total}$ with $\mathcal{L}_{val}, \mathcal{L}_{smooth}$ via Eq. (8)
6: Obtain updated actor model $\pi_{\Theta'_a}(\cdot)$ with loss $\mathcal{L}_{total}$
7: **return** $\pi_{\Theta'_a}(\cdot)$, a_t



(a) Entropy and critic value for GC-VAT on DAT. (b) Entropy and critic value for EVT on UnrealCV.

Fig. 2: Comparison of output entropy and critic value across tracking success/failure in out-of-distribution (OOD) and success in in-distribution (ID) scenarios. (a) GC-VAT [30] on DAT [30] benchmark. (b) EVT [46] on UnrealCV [27].

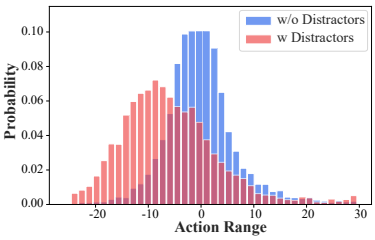


Fig. 3: Action distributions of EVT [46] in the same episode with and without distractors. Significant action shifts are observed when distractors appear.

distractor interference compared to idealized simulators. To address these issues, we propose **VAT** with Test-Time **A**daptation (**VATA**) method, which leverages test-time observations to update model parameters in an unsupervised manner. VATA aims to produce reliable tracking commands under these challenging domain shifts, thereby enhancing the tracker robustness. As shown in Fig. 1(b), VATA integrates two core components: 1) a **Critic Value Maximization** strategy to bridge visual domain shifts (Sec. 4.1), and 2) an **Action Smoothness Regularization** term to ensure reliability of output commands under distractors (Sec. 4.2). The pseudo-code of VATA is summarized in Alg. 1.

4.1 Test Time Adaptation with Critic Value Maximization

To address visual distribution shifts in unseen VAT environments, we propose a value-driven adaptation strategy based on two critical criteria: 1) the adaptation signal should serve as a discriminative metric to distinguish reliable test samples, and 2) the optimization objective should bridge the performance gap between in-domain and out-of-domain scenarios. Based on these criteria, we select the critic value as the adaptation signal instead of the commonly used entropy [23, 38], and employ value maximization to guide model updates.

Selecting Critic Value as Adaptation Signal. We evaluate the commonly used entropy and critic value metrics across multiple benchmarks (UnrealCV [27], DAT [30]) and methods (EVT [46], GC-VAT [30]). We compared these metrics during successful tracking versus target loss, where a larger gap indicates better discriminative power. As shown in Fig. 2, the model prediction entropy exhibits negligible differences between success and failure cases: only 3.4% on the DAT dataset (0.580 vs. 0.560) and 3.8% on UnrealCV (0.104 vs. 0.100). This indicates that the tracker becomes over-confident when losing the target, making entropy ineffective as a metric for distinguishing reliable samples. In contrast, the critic value demonstrates significantly superior discriminability. Specifically, on the DAT dataset, the critic value drops by 66% upon target loss (from 9.12 to 3.10). These results confirm that the critic value serves as a far more reliable indicator than output entropy for identifying trustworthy actions.

Maximizing Critic Value. Therefore, we can employ a threshold Q_0 on the critic value $Q(s_t, a_t)$ to filter reliable tracking actions for adaptation. As shown in Fig. 2, the critic assigns higher values to successful actions in-domain than in out-of-domain (OOD) scenarios, indicating a confidence drop under distribution shifts. To bridge this gap, we maximize the output values for reliable OOD actions, aligning the critic with in-domain standards. The objective functions for continuous and discrete tasks are defined in Eq. (2) and Eq. (3), respectively.

For continuous control tasks, we directly maximize the value of the output action $\pi_\theta(s_t)$, formulated as:

$$\mathcal{L}_{val}^{cont}(\theta) = -\mathbb{I}_{\{Q_t > Q_0\}}(Q_t) \cdot \mathbb{E}_{s_t}\left[Q\big(s_t, \pi_\theta(s_t)\big)\right], \tag{2}$$

where $\mathbb{I}_{\{Q_t > Q_0\}}(\cdot)$ is the indicator function, with Q_t denoting the current critic value $Q\big(s_t, \pi_\theta(s_t)\big)$. For discrete control tasks, we maximize the expected value over the current states with the following objective function:

$$\mathcal{L}_{val}^{disc}(\theta) = -\mathbb{I}_{\{Q_t > Q_0\}}(Q_t) \cdot \mathbb{E}_{s_t}\left[Q(s_t)\right], \tag{3}$$

where $Q_t = Q(s_t)$ denotes the potential value of the current state.

4.2 Handling Distractors with Action Smoothness Regularization

In real-world VAT scenarios, similar-looking distractors often cause the tracker to misidentify the target, leading to sudden action shifts as the agent abruptly steers toward the distractors, as shown in Fig. 3. Since real-world targets follow continuous motion dynamics, an optimal tracker should generate smooth actions consistent with these physical laws. To achieve this, we introduce an action smoothness regularization to penalize sudden jumps in control commands.

Theoretical Motivation of Action Regularization. For any target $\mathcal{T}$ in a real-world VAT scenario, its motion adheres to the laws of inertia. Specifically, we assume the target's position function $\mathbf{x}_\mathcal{T}(t)$ is differentiable w.r.t time t, and its velocity is constrained by a physical upper bound v_{max}, such that $\|\dot{\mathbf{x}}_\mathcal{T}(t)\| \leq v_{max}$. Furthermore, we define an optimal VAT tracker π^* that maps the target's state to the optimal control command at any time t: $a^*(t) = \pi^*(\mathbf{x}_\mathcal{T}(t))$.

Proposition 1 *Under the assumptions that the optimal tracking policy π^* is locally Lipschitz continuous with a Lipschitz constant L, the time derivative of the optimal action $a^*(t)$ is bounded for any timestep t:*

$$|\dot{a}^*(t)| \leq L \cdot v_{max}. \tag{4}$$

For the proof of Proposition 1, please refer to the Appendix. Proposition 1 demonstrates that the variation in tracker actions is upper-bounded. Based on this, we can identify actions exceeding this limit and classify them as irrational control signals induced by distractors. Consequently, we apply action smoothness regularization to penalize such deviations, enabling the tracker to maintain robust locking on the correct target over time.

Action Smoothness Regularization. Since distractor interference primarily causes abrupt orientation shifts, we apply smoothness regularization to the rotation component of the action, denoted as r_t. We identify the control jumps by measuring the deviation of the current rotation command from the historical movement trend. Specifically, we maintain a sliding buffer $\mathcal{H}_t = \{r_{t-K}, \ldots, r_{t-1}\}$ consisting of the rotation actions from the previous K steps. The historical trend is then represented by the moving average of the buffer: $\bar{r}_t = \frac{1}{K}\sum_{i=1}^{K} r_{t-i}$. Therefore, we can quantify the jumps in control commands as:

$$d_t = \frac{r_t - \bar{r}_t}{\sigma}, \tag{5}$$

where σ denotes a scaling factor that normalizes the action range. When the action jump d_t exceeds a predefined threshold δ, we consider it as distractor interference and apply regularization to the action as follows:

$$\mathcal{L}_{smooth}(\theta) = \left\|\texttt{ReLU}(|d_t| - \delta)\right\|_2^2, \tag{6}$$

where $\texttt{ReLU}(\cdot) = \max(0, \cdot)$ ensures regularization $\mathcal{L}_{smooth}$ is activated only when the deviation d_t surpasses the threshold δ.

Remark on Extensibility of our Regularization. We should highlight that the proposed action smoothness regularization is not limited to reinforcement learning(RL)-based VAT trackers. It is equally applicable to Imitation Learning (IL)-based trackers, such as TrackVLA [33]. Consider an IL-based tracker $\mathcal{T}_{IL}$ that takes images as input and predicts a future trajectory of N points, denoted as $\mathcal{P} = \{p_1, \ldots, p_N\}$ with $p_i \in \mathbb{R}^3$. The tracker then executes the first M actions ($M < N$) from this plan before triggering the next planning.

Similar to our RL formulation, we measure action jumps at step $t+M$ by comparing the newly planned trajectory $\{{p'}^1_{t+M}, \ldots, {p'}^N_{t+M}\}$ at step $t+M$ with the remaining future path from the previous plan $\{p_t^{M+1}, \ldots, p_t^{M+K}\}$ at step t. The jumps in control commands are quantified as:

$$d_{t+M} = \frac{1}{N-M}\sum_{k=1}^{N-M} \frac{\|{p'}^k_{t+M} - p_t^{M+k}\|_2}{\sigma_{IL}}, \tag{7}$$

where σ_{IL} denotes a normalization factor. Consequently, the same objective function defined in Eq. (6) can be directly employed to regularize the model, ensuring smoothness of the tracker trajectories under similar looking distractors.

Table 1: Comparison results in UnrealCV environment with distractors. **Bold** represents the best while underline represents the second.

Method	Parking Lot (2D)			UrbanCity (4D)			ComplexRoom (4D)			Average		
	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑
EVT	220	420	0.64	335	472	0.86	321	465	0.86	292	452	0.79
TENT	172	369	0.48	270	419	0.76	302	449	0.74	248	412	0.66
EATA	272	443	0.66	347	471	0.88	329	472	0.86	316	462	0.80
COME	280	445	0.68	349	473	0.89	333	473	0.88	321	464	0.82
TARL	289	446	0.70	337	**474**	0.88	334	477	0.88	320	466	0.82
Ours	**302**	**449**	**0.74**	**350**	**474**	**0.90**	**355**	**480**	**0.92**	**336**	**468**	**0.85**

Total Optimization Objective. The final test-time adaptation objective combines our value maximization goal with the action smoothness constraint:

$$\mathcal{L}_{total} = \lambda_v \mathcal{L}_{val} + \lambda_s \mathcal{L}_{smooth}, \tag{8}$$

where λ_v and λ_s are balancing coefficients. Note that the action smoothness regularization $\mathcal{L}_{smooth}$ is applied only to continuous tracking tasks, and we set $\lambda_s = 0$ for discrete tasks. During the model adaptation, we only update the affine parameters (scale and shift) of the batch normalization layers. This helps the model adjust to new environments without losing the task knowledge learned during the pre-training phase.

5 Experiments

5.1 Experimental Setup

Environments. We evaluate VATA on three state-of-the-art VAT benchmarks.

1) **UnrealCV** [27]: An indoor ground-to-ground tracking simulator built on Unreal Engine 4 [12]. It provides high-quality visual rendering and complex object layouts. We train our model on samples collected in the *Complex Room* environment, and evaluate its transfer performance in *Simple Room (2D)*, *Parking Lot (4D)*, and *Urban City (4D)* with varying numbers of distractors.

2) **DAT** [30]: An air-to-ground tracking benchmark built on Webots [19], featuring diverse outdoor scenes and complex agent behaviors. We train the tracker in *citystreet* environment under *daytime* conditions. To evaluate out-of-distribution (OOD) generalization, we test the tracker on domain-shifted variants of *citystreet* with altered weather (*foggy*, *night*, *snow*), as well as on scene-shifted environments (*desert*, *downtown*, *lake*, and *farmland*) in *daytime*.

3) **EVT-Bench** [33]: A large-scale ground-based tracking simulator built on Habitat [26], featuring highly challenging conditions such as severe occlusions. The benchmark consists of three tasks. *Single Target Tracking (STT)*: the agent tracks a single target in the map; *Distracted Tracking (DT)*: the tracker should lock on the target under distractors; and *Ambiguity Tracking (AT)*: the tracker

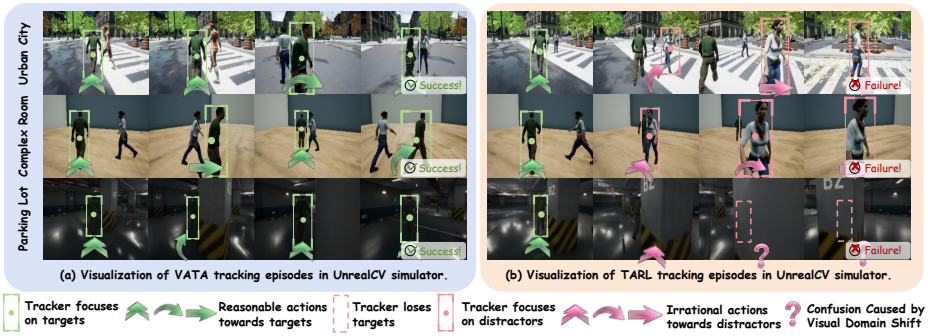


Fig. 4: Qualitative results on the UnrealCV benchmark. (a) Visualization of tracking episodes using VATA. (b) Visualization of TARL tracking episodes.

is required to track the target among multiple candidates with identical appearances. We train our model on samples collected from the *STT* training scenes and evaluate its OOD performance across all three tasks on the test scenes.

Metrics. In **UnrealCV**, we report three metrics: *Accumulated Reward (AR)* measures the average reward over 100 episodes. *Episode Length (EL)* denotes the average steps per episode, with early termination if the target remains out of view for more than 50 consecutive steps. *Success Rate (SR)* represents the proportion of success episodes. In **DAT**, *Cumulative Reward (CR)* measures the average total reward over 40 episodes, and *Tracking Success Rate (TSR)* denotes the success rate within 1,500 steps. In **EVT-Bench**, *Success Rate (SR)* measures the proportion of successful episodes out of all episodes. *Tracking Rate (TR)* measures the proportion of successful tracking steps out of all steps. *Collision Rate (CR)* measures the proportion of episodes terminated due to collisions between the tracker and the environment.

Baselines. We select state-of-the-art RL-based VAT trackers trained on different simulators as our base models: EVT [46] for UnrealCV and EVT-Bench, and GC-VAT [30] for DAT. Moreover, we adopt four recent test-time adaptation (TTA) methods as baselines: 1) **TENT** [32] minimizes policy output entropy to reduce prediction uncertainty. 2) **EATA** [23] first filters high-confidence samples based on the model's output entropy, and adapts the model at test time by minimizing prediction entropy. It further employs Fisher regularization to mitigate catastrophic forgetting. 3) **COME** [41] leverages a Dirichlet prior for uncertainty modeling to resolve the overconfidence and instability of entropy minimization; 4) **TARL** [38] employs entropy-based sample selection and action entropy-minimization for offline RL policies.

Implementation Details. To stabilize optimization, we first run a 50-step warm-up phase before test-time adaptation. We then update the parameters of the batch normalization layers every 5 steps with learning rate $\eta = 5 \times 10^{-4}$. The critic value threshold in Eq. (2) is set to $Q_0 = 10$. For action smoothness regularization, we use an action jump threshold $\delta = 0.2$ in Eq. (6) and a history window size $K = 10$. The total loss weights are $\lambda_v = 0.03$ and $\lambda_s = 10$ in Eq. (8).

Table 2: Experimental results on the DAT benchmark. We report the Cumulative Reward (*CR*) and Tracking Success Rate (*TSR*) in the format of *CR*/*TSR*.

Method	Cross-Domain			Cross-Scene			
	Foggy	Night	Snow	Desert	Downtown	Lake	Farmland
GC-VAT	265/0.74	199/0.60	152/0.52	119/0.56	148/0.50	139/0.45	138/0.44
TENT	276/0.76	188/0.57	145/0.48	103/0.52	158/0.52	138/0.44	150/0.47
EATA	300/0.77	218/0.65	149/0.49	106/0.53	160/0.54	168/0.49	175/0.52
COME	306/0.78	220/0.66	150/0.49	111/0.55	161/0.54	170/0.50	180/0.54
TARL	310/0.79	203/0.62	160/0.52	129/0.58	158/0.53	152/0.47	183/0.55
Ours	**316/0.80**	**224/0.67**	**165/0.53**	**140/0.59**	**170/0.55**	**174/0.51**	**212/0.58**

5.2 Comparison Experiments

Effectiveness in UnrealCV Environments with Distractors. We evaluate VATA in UnrealCV maps with similar-looking distractors. As shown in Tab. 1, VATA achieves a relative Success Rate (SR) improvement of 7.6% over the base tracker EVT [46] (0.79 → 0.85) and outperforms the SOTA TTA method TARL by 3.7% (0.82 → 0.85) across three scenarios. These results demonstrate the efficacy of our approach in enhancing VAT tracker performance during test time. To further illustrate these gains, we present qualitative comparisons between TARL and VATA. As shown in Row 3 of Fig. 4, our method maintains robust tracking in a low-light scenario while TARL fails. Furthermore, in scenarios with visually similar distractors (Row 1-2 of Fig. 4), VATA maintains accurate target tracking, while TARL is misled and tracks the distractors.

Effectiveness in DAT Environments. As shown in Tab. 2, VATA demonstrates consistent improvements across both cross-domain and cross-scene evaluations on DAT. Specifically, our method outperforms the base tracker GC-VAT by 8.1% in average success rate (0.62 → 0.67). Furthermore, it surpasses the TARL method by 4.7% (0.64 → 0.67) with a notable gain of 5.7% (0.53 → 0.56) in the challenging cross-scene setting. Given that DAT employs a discrete action space, VATA only leverages the critic value maximization strategy defined in Eq. (3). These results show the superiority of the proposed critic value maximization over conventional entropy minimization schemes for VAT tasks.

Effectiveness in EVT-Bench Environments. As shown in Tab. 3, VATA achieves SOTA performance across three tasks. Specifically, our method outperforms the base tracker by an average of 48.3% in SR across all tasks (15.1% → 22.4%) and surpasses the second-best TTA baseline by 24.4% (18.0% → 22.4%) on average. Notably, the most significant gains are observed in the *DT* and *AT* tasks, which contain visually similar distractors. Compared to the runner-up, VATA yields 64.7% improvement in SR for *DT* (6.8% → 11.2%) and a 22.1% gain for *AT* (20.8% → 25.4%). Further qualitative evidence is provided in Fig. 5.

Table 3: Comparison results on EVT-Bench. All metrics are reported in percentages. **Bold** represents the best while underline represents the second.

Method	STT			DT			AT			Average		
	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓
EVT	24.3	40.2	42.1	3.4	12.3	47.1	17.6	21.7	46.2	15.1	24.7	45.1
TENT	25.2	39.5	40.8	4.5	13.5	44.5	18.5	23.1	46.9	16.1	25.4	44.1
EATA	25.8	40.6	41.9	5.2	14.8	45.8	19.4	24.2	45.9	16.8	26.5	44.5
COME	26.2	41.7	41.6	5.9	15.8	44.2	20.8	25.7	45.2	17.6	27.7	43.7
TARL	26.7	42.4	41.7	6.8	16.5	46.3	20.6	25.8	46.1	18.0	28.2	44.7
Ours	**30.5**	**45.6**	**39.0**	**11.2**	**23.6**	**42.2**	**25.4**	**28.4**	**41.0**	**22.4**	**32.5**	**40.7**

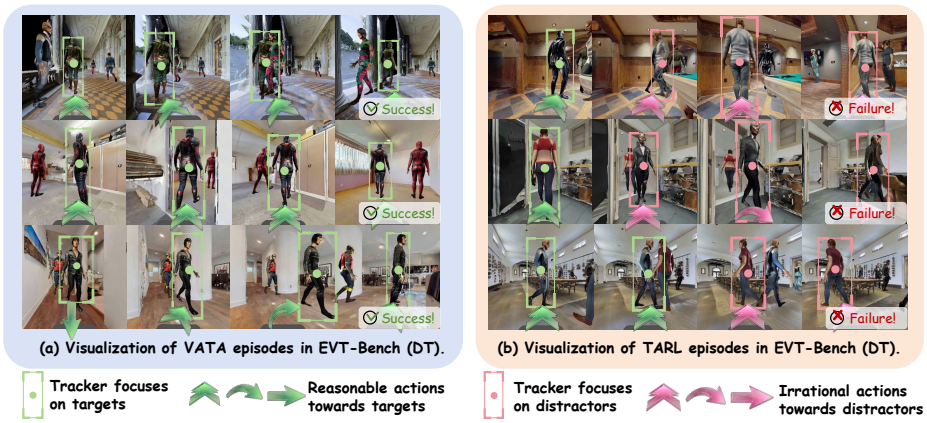


Fig. 5: Qualitative results on the EVT benchmark. (a) Visualization of tracking episodes using VATA. (b) Visualization of TARL tracking episodes.

5.3 Ablation Experiments

We validate the effectiveness of the proposed Critic Value Maximization strategy (Sec. 4.1) and Action Smoothness Regularization (Sec. 4.2) on UnrealCV maps. Furthermore, we conduct hyperparameter experiments on UnrealCV to verify the method's stability across different settings. Additionally, to demonstrate the extensibility of the proposed Action Smoothness Regularization to imitation-learning-based approaches, we evaluate TrackVLA [33] method on EVT-Bench.

Effectiveness of VATA modules. As shown in Row 2 of Tab. 4, when VATA runs without Critic Value Maximization, the tracking success rate drops by 5.9% (from 0.85 to 0.80). Furthermore, when our method operates without Action Smoothness Regularization, performance similarly declines, showing a 3.5% decrease in SR (from 0.85 to 0.82), as detailed in Row 3 of Tab. 4. These results confirm the effectiveness of both modules.

Robustness to Hyperparameters. We conduct hyperparameter experiments on the critic value threshold Q_0 in Eq. (2) and the action jump threshold σ in Eq. (6) to verify hyperparameter stability. As shown in Tab. 5, we compare different parameter combinations with $Q_0 \in [5, 10, 20]$ and $\delta \in [0.1, 0.2, 0.3]$. The

Table 4: Ablation study of VATA on UnrealCV benchmark. "w/o $\mathcal{L}_{val}$" denotes removing the critic value maximization strategy, and "w/o $\mathcal{L}_{smooth}$" denotes removing the action smoothness regularization term.

Version	Parking Lot (2D)			UrbanCity (4D)			ComplexRoom (4D)			Average		
	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑
EVT	220	420	0.64	335	472	0.86	321	465	0.86	292	452	0.79
w/o $\mathcal{L}_{val}$	235	435	0.66	338	473	0.87	325	468	0.87	299	459	0.80
w/o $\mathcal{L}_{smooth}$	280	435	0.70	345	472	0.88	345	472	0.89	323	460	0.82
Ours	**302**	**449**	**0.74**	**350**	**474**	**0.90**	**355**	**480**	**0.92**	**336**	**466**	**0.85**

Table 5: Sensitivity analysis of hyperparameters Q_0 and δ. We report the detailed performance across three UnrealCV environments.

Params		Parking Lot (2D)			UrbanCity (4D)			ComplexRoom (4D)			Average		
Q_0	δ	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑	AR ↑	EL ↑	SR ↑
5	0.1	298	442	0.71	348	468	0.88	353	473	0.90	333	461	0.83
10	0.1	295	440	0.72	346	465	0.89	349	472	0.91	330	459	0.84
20	0.2	296	446	0.73	344	470	0.89	347	476	0.91	329	464	0.84
5	0.3	290	435	0.69	340	462	0.87	345	468	0.90	325	455	0.82
10	0.3	300	445	0.73	348	472	**0.90**	354	469	0.91	334	462	**0.85**
10	**0.2**	**302**	**449**	**0.74**	**350**	**474**	**0.90**	**355**	**480**	**0.92**	**336**	**466**	**0.85**

variation in the AR metric across these combinations is less than 3.3%, demonstrating that VATA maintains high stability regardless of parameter choices.

Extensibility of Action Smoothness Regularization. To validate the applicability of our Action Smoothness Regularization to imitation learning-based VAT trackers, we evaluate it on the TrackVLA [33] method within EVT-Bench, as detailed in Sec. 4.2. Since the original TrackVLA is not open-source, we conduct experiments using its publicly available version, Open-TrackVLA [16]. Specifically, we train the model on data collected from the STT training environments and evaluate its performance under DT task. Furthermore, given that TrackVLA employs Layer Normalization (LN) instead of Batch Normalization (BN), we adapt our test-time adaptation strategy to update the LN parameters. As shown in Tab. 6, integrating our action regularization yields a 45.7% improvement in SR on the DT task (11.6% → 16.9%). These results confirm the effectiveness and extensibility of our regularization.

5.4 Experiments in Real-world Scenarios

We validate the effectiveness of VATA for real-world VAT deployment on a DJI Tello drone [7]. During operation, the drone streams H.264 video to a ground station with an NVIDIA RTX 3090 GPU over Wi-Fi. The ground station decodes the video stream and executes the tracker model to generate control commands, while applying VATA for online model adaptation. These commands are then

Table 6: Comparison results between Open-TrackVLA and Open-TrackVLA with our Action Smoothness Regularization.

Method	DT		
	SR ↑	TR ↑	CR ↓
Open-TrackVLA	11.6	47.8	4.8
Ours	**16.9**	**50.6**	**4.2**

Table 7: Experimental results on a physical drone.

Method	Real Drone
	TSR ↑
EVT	0.2
TARL	0.3
Ours	**0.8**

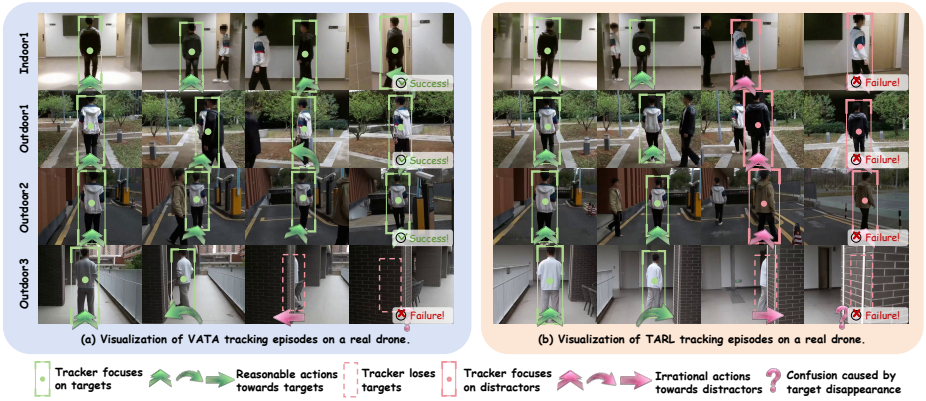


Fig. 6: Qualitative results on real robot deployment. (a) Visualization of tracking episodes using VATA. (b) Visualization of TARL tracking episodes.

transmitted back to the drone, enabling closed-loop tracking. For these experiments, we employ an EVT model trained on UnrealCV as the base tracker.

We evaluate VATA across 10 episodes in diverse indoor and outdoor environments, all exhibiting significant visual domain shifts compared to the UnrealCV maps. Moreover, each episode includes human distractors to test the model's robustness against interference. We evaluate performance using the Tracking Success Rate (*TSR*), which is defined as the fraction of fully successful episodes.

Effectiveness on Real Robots. As shown in Tab. 7, the base EVT tracker succeeds in only 2 out of 10 episodes. In contrast, VATA achieves an 80% TSR, outperforming the SOTA method TARL (30%) and demonstrating superior real-world robustness. We further provide episode visualizations in Fig. 6. When human distractors appear (Fig. 6 Row 1-3), EVT tracks the incorrect target, whereas VATA maintains lock on the correct target. Moreover, the entire system runs at 30 FPS, enabling real-time tracking. See more results in Appendix.

Limitation of VATA. As shown in Row 4 of Fig. 6, VATA fails to recover tracking after prolonged occlusion. This is mainly because existing methods lack trajectory planning based on historical memory. Such planning remains particularly challenging in unsupervised TTA paradigms due to absence of external supervision signals. We will explore it in the future work.

6 Conclusion

In this paper, we propose a test-time adaptation method for VAT called VATA, which enables superior tracking performance in the real world. Specifically, we introduce a **Critic Value Maximization** strategy that uses the critic value of the RL-based VAT tracker to filter reliable test samples, and guides model updates under visual domain shift. Moreover, we propose an **Action Smoothness Regularization** term to correct tracker actions under similar-looking distractors, ensuring the tracker locks onto the correct target. Experiments on simulators and a real drone demonstrate VATA's state-of-the-art performance and effectiveness in real-world VAT deployment.

Acknowledgements

This work was partially supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U24A20327) and Guangdong S&T Program under Grant 2026B0101110001.

References

1. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3674–3683 (2018)
2. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: 2009 IEEE Conference on computer vision and Pattern Recognition. pp. 983–990. IEEE (2009)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016)
4. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6182–6191 (2019)
5. Chen, X., Peng, H., Wang, D., Lu, H., Hu, H.: Seqtrack: Sequence to sequence learning for visual object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14572–14581 (June 2023)
6. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), `https://lmsys.org/blog/2023-03-30-vicuna/`
7. Da-Jiang Innovations: Dji tello. https://store.dji.com/product/tello (2025)
8. Devo, A., Dionigi, A., Costante, G.: Enhancing continuous control of mobile robots for end-to-end visual active tracking. Robotics and Autonomous Systems **142**, 103799 (2021)
9. Dionigi, A., Felicioni, S., Leomanni, M., Costante, G.: D-vat: End-to-end visual active tracking for micro aerial vehicles. IEEE Robotics and Automation Letters **9**(6), 5046–5053 (2024)
10. Emran, B.J., Najjaran, H.: A review of quadrotor: An underactuated mechanical system. Annual Reviews in Control **46**, 165–180 (2018)
11. Fan, Y., Chen, W., Jiang, T., Zhou, C., Zhang, Y., Wang, X.E.: Aerial vision-and-dialog navigation. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 3043–3061. Association for Computational Linguistics, Toronto, Canada (Jul 2023). `https://doi.org/10.18653/v1/2023.findings-acl.190`
12. Games, E.: Unreal engine 4. https://www.unrealengine.com/ (2025), software
13. Gao, J., Yao, X., Xu, C.: Fast-slow test-time adaptation for online vision-and-language navigation. arXiv preprint arXiv:2311.13209 (2023)
14. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
15. Ko, H., Kim, S., Oh, G., Yoon, J., Lee, H., Jang, S., Kim, S., Kim, S.: Active test-time vision-language navigation. arXiv preprint arXiv:2506.06630 (2025)
16. Lee, K., Zhao, T.: Opentrackvla: Open-source visual language action model for visual navigation and following. https://github.com/om-ai-lab/OpenTrackVLA (2025)
17. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8971–8980 (2018)

18. Liu, S., Zhang, H., Qi, Y., Wang, P., Zhang, Y., Wu, Q.: Aerialvln: Vision-and-language navigation for uavs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15384–15394 (2023)
19. Ltd., C.: Webots: Open source mobile robot simulation software. https://cyberbotics.com/ (2025), software
20. Luo, W., Sun, P., Zhong, F., Liu, W., Zhang, T., Wang, Y.: End-to-end active object tracking and its real-world deployment via reinforcement learning. IEEE transactions on pattern analysis and machine intelligence **42**(6), 1317–1332 (2019)
21. Maalouf, A., Jadhav, N., Jatavallabhula, K.M., Chahine, M., Vogt, D.M., Wood, R.J., Torralba, A., Rus, D.: Follow anything: Open-set detection, tracking, and following in real-time. IEEE Robotics and Automation Letters **9**(4), 3283–3290 (2024)
22. Minorsky, N.: Directional stability of automatically steered bodies. Journal of the American Society for Naval Engineers **34**(2), 280–309 (1922)
23. Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., Tan, M.: Efficient test-time model adaptation without forgetting. In: International conference on machine learning. pp. 16888–16905. PMLR (2022)
24. Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., Tan, M.: Towards stable test-time adaptation in dynamic wild world. In: International Conference on Learning Representations (ICLR) (2023)
25. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023)
26. Puig, X., Undersander, E., Szot, A., Cote, M.D., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y., et al.: Habitat 3.0: A co-habitat for humans, avatars and robots. arXiv preprint arXiv:2310.13724 (2023)
27. Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T.S., Wang, Y.: Unrealcv: Virtual worlds for computer vision. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1221–1224 (2017)
28. Schedl, D.C., Kurmi, I., Bimber, O.: An autonomous drone for search and rescue in forests using airborne optical sectioning. Science Robotics **6** (2021)
29. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1442–1468 (2013)
30. Sun, H., Hu, J., Zhang, Z., Tian, H., Xie, X., Wang, Y., Xie, X., Lin, Y., Yu, Z., Tan, M.: Open-world drone active tracking with goal-centered rewards. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
31. Tan, M., Chen, P., Zhi, H., Mai, J., Rosman, B., Ji, D., Zeng, R.: Source-free elastic model adaptation for vision-and-language navigation. IEEE Transactions on Multimedia (2025)
32. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)
33. Wang, S., Zhang, J., Li, M., Liu, J., Li, A., Wu, K., Zhong, F., Yu, J., Zhang, Z., Wang, H.: Trackvla: Embodied visual tracking in the wild (2025), `https://arxiv.org/abs/2505.23189`
34. Wen, L., Du, D., Zhu, P., Hu, Q., Wang, Q., Bo, L., Lyu, S.: Detection, tracking, and counting meets drones in crowds: A benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7812–7821 (2021)

35. Wu, K., Xu, S., Chen, H., Wang, C., Li, Z., Wang, Y., Zhong, F.: Vlm can be a good assistant: Enhancing embodied visual tracking with self-improving vision-language models. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13154–13161. IEEE (2025)
36. Wu, Y., Wang, X., Yang, X., Liu, M., Zeng, D., Ye, H., Li, S.: Learning occlusion-robust vision transformers for real-time uav tracking. In: CVPR (2025)
37. Xing, L., Fan, X., Dong, Y., Xiong, Z., Xing, L., Yang, Y., Bai, H., Zhou, C.: Multi-uav cooperative system for search and rescue based on yolov5. International Journal of Disaster Risk Reduction **76**, 102972 (2022)
38. Xu, S., Tan, M., Liu, L., Zhang, Z., Zhao, P., et al.: Test-time adapted reinforcement learning with action entropy regularization. In: Forty-second International Conference on Machine Learning
39. Ye, B., Chang, H., Ma, B., Shan, S., Chen, X.: Joint feature learning and relation modeling for tracking: A one-stream framework. In: European Conference on Computer Vision. pp. 341–357. Springer (2022)
40. Zhang, C., Zhou, W., Qin, W., Tang, W.: A novel uav path planning approach: Heuristic crossing search and rescue optimization algorithm. Expert Systems with Applications **215**, 119243 (2023)
41. Zhang, Q., Bian, Y., Kong, X., Zhao, P., Zhang, C.: Come: Test-time adaption by conservatively minimizing entropy. In: The Thirteenth International Conference on Learning Representations
42. Zhi, H., Chen, P., Li, J., Ma, S., Sun, X., Xiang, T., Lei, Y., Tan, M., Gan, C.: Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 3761–3771 (2025)
43. Zhong, F., Bi, X., Zhang, Y., Zhang, W., Wang, Y.: Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 3705–3714 (2023)
44. Zhong, F., Sun, P., Luo, W., Yan, T., Wang, Y.: Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. IEEE transactions on pattern analysis and machine intelligence **43**(5), 1467–1482 (2019)
45. Zhong, F., Sun, P., Luo, W., Yan, T., Wang, Y.: Ad-vat: An asymmetric dueling mechanism for learning visual active tracking. In: International Conference on Learning Representations (2019)
46. Zhong, F., Wu, K., Ci, H., Wang, C., Chen, H.: Empowering embodied visual tracking with visual foundation models and offline rl. In: European Conference on Computer Vision. pp. 139–155. Springer (2024)
47. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H.: Detection and tracking meet drones challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 7380–7399 (2022). https://doi.org/10.1109/TPAMI.2021.3119563